ProxUSB (módulo OEM)

Conversor de protocolo wiegand a interfaz USB para lectores de RFID



Manual del usuario

(Revisión 2.0)

Contenido

Descripción del módulo

El módulo ProxUSB permite conectar cualquier lector de tarjetas de proximidad con salida wiegand a una PC o laptop con interfaz USB. De esta forma se puede incorporar de forma práctica y sencilla la tecnología RFID en aplicaciones que requieran el uso de tarjetas de proximidad para identificación de visitas, empleados, procesos de producción, etc.

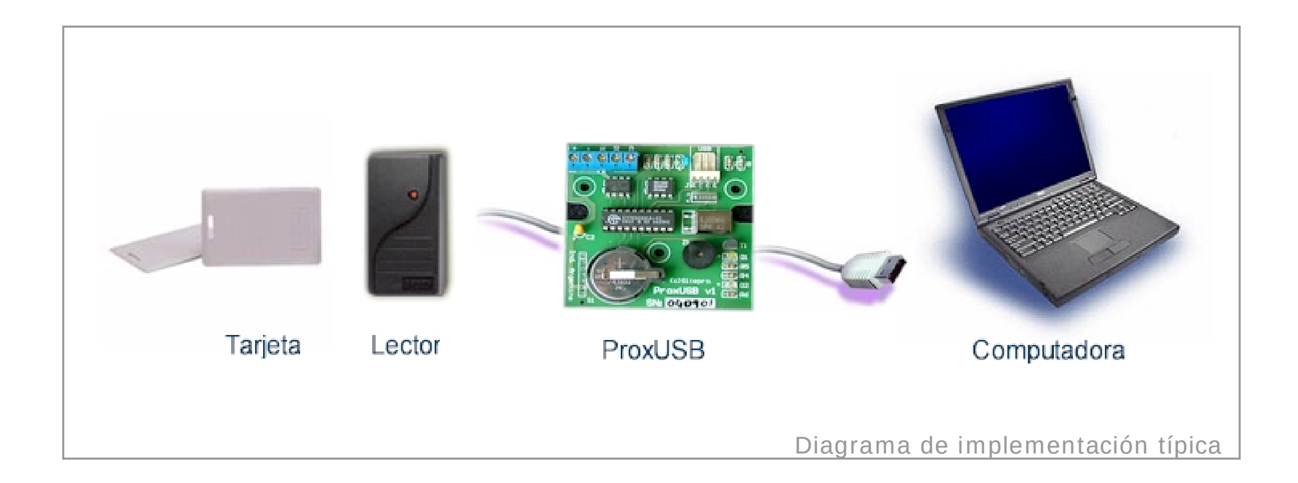
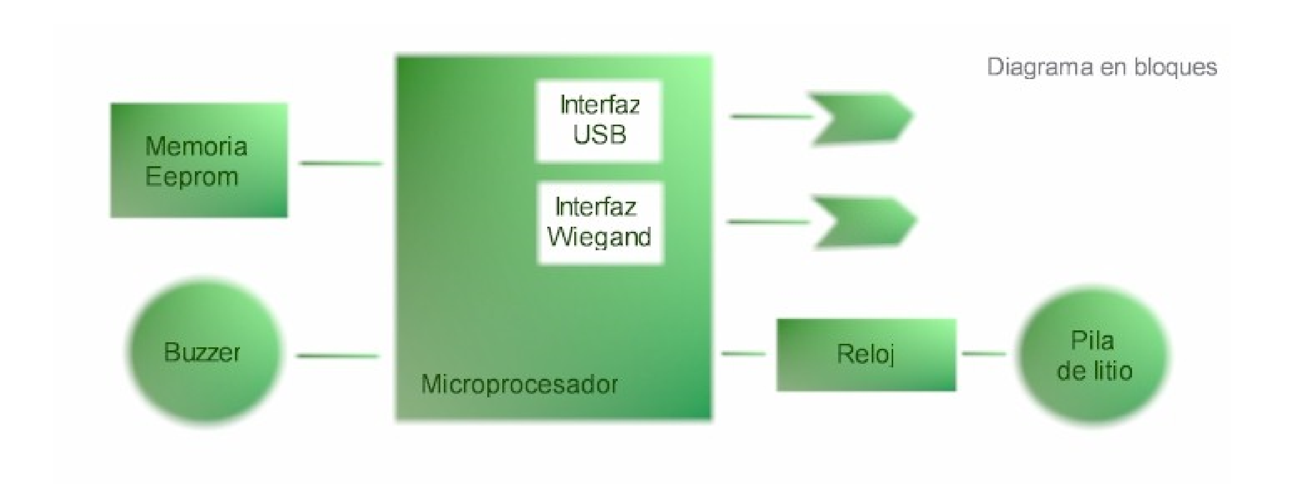


Diagrama de implementación típica

El protocolo soportado para lectores de tarjetas de proximidad es wiegand 26 bits Standard. Esto asegura compatibilidad con la gran mayoría de implementaciones de RFID. La tensión del módulo y del lector de tarjetas se toma de la propia interfaz USB con lo cual no se requiere el uso de fuentes de alimentación externas (ver características técnicas para más detalle de consumos admitidos).
El módulo contiene además un reloj de tiempo real y una memoria no volátil para el desarrollador que pueden ser accedidos a través de una librería para Windows.



La configuración del reloj se realiza mediante un utilitario protegido por una password que el desarrollador puede configurar según su criterio. De esta forma no puede ser accedida por el usuario final, lo cual es muy útil en aplicaciones de control, auditoria, tiempo y asistencia, etc.

Descripción del utilitario de configuración

El utilitario ProxUSBconfig.exe permite ver y/o cambiar la hora y fecha que maneja el reloj interno de la interfase. En el primer campo del utilitario se debe ingresar el password de acceso que permite grabar los cambios al poner en hora y fecha el reloj. Este password también es requerido para la función que permite grabar un password nuevo. La función que permite leer la configuración del reloj no requiere password alguno.



La **password** que viene **pre-configurada** de fábrica son ocho ceros **"00000000"**. Se puede usar hasta ocho caracteres, números, símbolos o letras. Se debe tener presente que las letras en mayúsculas y minúsculas se asumen como caracteres distintos.

Si su aplicación es sensible al cambio de hora y fecha por parte de un usuario final, es recomendable cambiar la password de fábrica por una propia antes de entregar el producto.

También esta previsto en el futuro la posibilidad de contemplar más formatos en el protocolo de comunicación del lector, lo cual se podrá configurar mediante este utilitario.

Descripción del utilitario emulador de teclado

El utilitario ProxUSBtec.exe es un pequeño programa residente que se encarga de monitorear la lectura de tarjetas y presentarle al sistema operativo los datos leídos como si hubieran ingresado por el teclado. Esto es muy útil para integrar de forma inmediata la captura de datos en aplicaciones existentes sin necesidad de modificar el código fuente.

Una vez ejecutado el programa se visualiza como un pequeño icono en la barra de tareas del Windows, junto a la esquina donde se muestra la hora de la PC.



Haciendo click con el botón derecho del mouse sobre el icono se accede a un menú de configuración donde es posible configurar la forma en que se presentan los datos a la aplicación. La siguiente pantalla muestra el aspecto de la ventana de configuración.

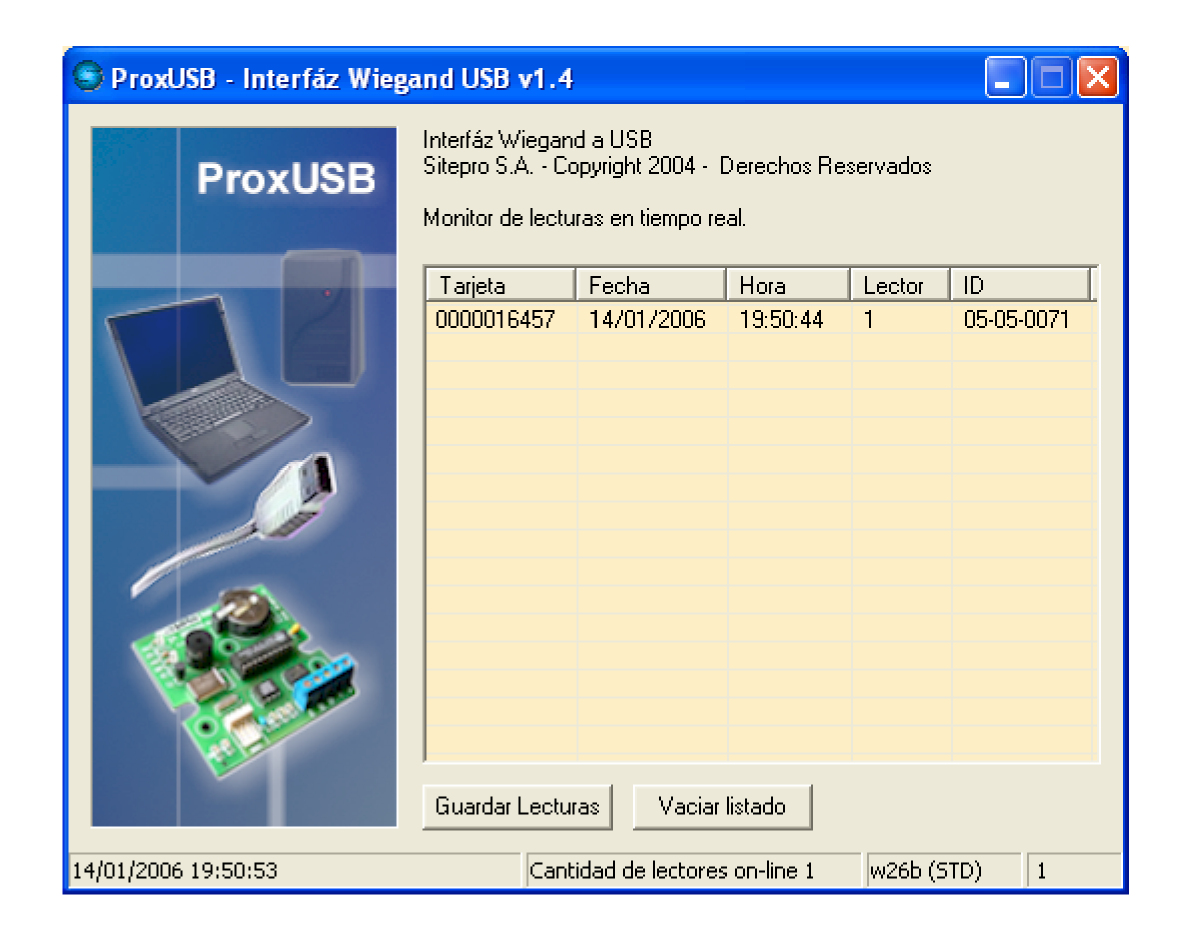


Desde aquí se puede configurar básicamente la forma en que el emulador presenta la información al buffer de teclado. Como se puede apreciar delante y atrás del campo con el número de tarjeta hay dos cuadros donde se pueden agregar hasta diez caracteres numéricos fijos que agregarán al número leído. Esto permite adaptarse fácilmente a cualquier aplicación existente donde se requiera completar con ceros para que el número generado tenga un largo determinado.

Atención: el emulador pone la información leída en la cola de mensajes de teclado del Windows, o sea que dichos caracteres ingresaran en el campo de la aplicación que este activa y tenga el foco del Windows en ese momento.

Descripción del programa demo ProxUSB.EXE

El programa ProxUSB.EXE es una pequeña demostración del potencial de la placa. Se trata de un ejecutable residente que permanece activo en background interrogando cada 2 segundos a la interfaz. Si se detecta que hubo una lectura de tarjeta satisfactoria, el programa obtiene la hora y fecha del RTC propio y va almacenando dichos datos en un buffer que puede consultarse por pantalla.



El botón “Guardar Lecturas” abre una ventana interrogando por un nombre de archivo donde salvar la información almacenada en el buffer de lecturas.

El botón “Vaciar listado” sirve para borrar toda la información acumulada en el buffer de lecturas y dejar el listado en blanco.

En la barra de estado inferior se puede ver la hora y fecha del reloj interno de la placa, el estado del lector y la cantidad de lecturas almacenadas en el buffer.

Atención: el buffer de lecturas se arma en forma dinámica en al memoria RAM de la PC. Dicha información se pierde si se sale del programa sin usar la opción “Guardar Lecturas” o si la PC se apaga abruptamente.

Detalle de conexiones

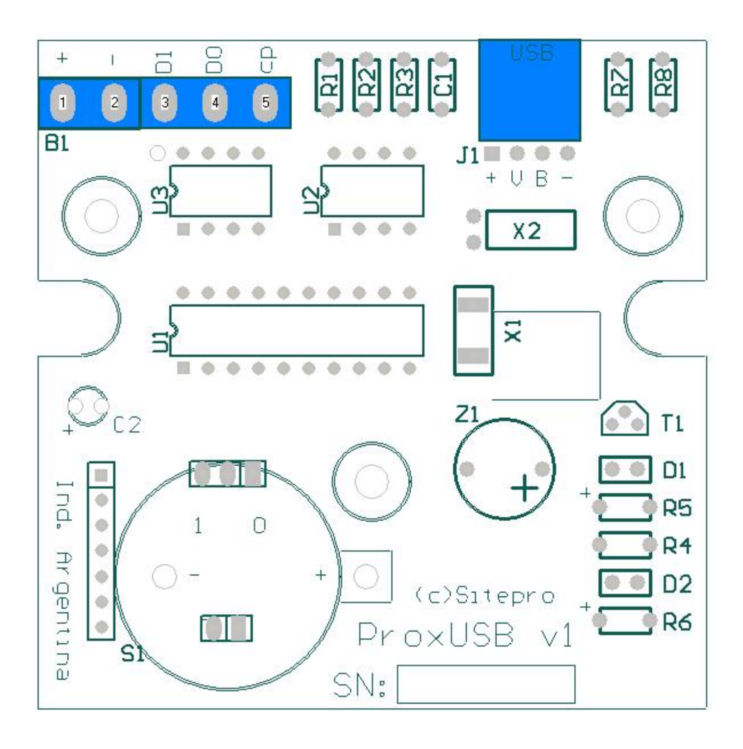
El módulo ProxUSB tiene una bornera que sirve para realizar la conexión del lector de tarjetas y una ficha polarizada para la conexión del cable USB que va a la computadora. A continuación se detalla las señales disponibles en cada conector:

Bornera B1 (conexión del lector de tarjetas):

+	Tensión de 5 volts para alimentar el lector de tarjetas de proximidad. Esta tensión se obtiene de la propia interfaz USB por lo tanto verifique que la interfaz utilizada sea capaz de entregar la corriente que requiere el lector. El consumo del lector de proximidad no deberá superar los 80 mA en ningún caso.
-	Señal de masa para conectar el lector de tarjetas.
D0	Señal Data0 del lector de proximidad.
D1	Señal Data1 del lector de proximidad.
Cp	Reservado.

Ficha J1 (conexión USB a la computadora):

+	Tensión de 5 volts.	(Rojo)
V	Señal D+ de la interfaz USB.	(Verde)
B	Señal D- de la interfaz USB.	(Blanco)
-	Señal de masa.	(Negro)



Detalle de la librería ProxUSB.DLL

El módulo ProxUSB puede manejarse desde una aplicación propia para Windows a través de una librería que es compatible con cualquier lenguaje de programación que soporte llamadas a una DLL.

La librería Prox-USB.DLL exporta una serie de funciones que permiten:

- Obtener información del módulo
- Consultar si fue leída alguna tarjeta
- Obtener los datos de la tarjeta de proximidad leída
- Consultar o actualizar la hora y fecha del reloj de tiempo real del módulo
- Leer o grabar información en la memoria no-volátil propia del módulo

En el CD encontrara un ejemplo en VisualBasic 6.0 de cómo declarar y utilizar las funciones de esta librería para obtener la información del reloj y la lectura de tarjetas. También se incluye un demo en VisulaC++ 6.0 donde se detalla como usar cada una de las funciones.

A continuación se detalla la sintaxis y los parámetros específicos de cada función en formato VisualC:

BuscarLector()

Esta función busca un módulo Prox-USB válido que esté conectado a algún puerto USB de la PC. Es la primera función que debe invocarse para verificar que haya algún lector conectado y a su vez dejar iniciada una conexión para luego leer las tarjetas.

Formato:

unsigned long int BuscarLector(char* num_serie)

Parámetros:

char* num_serie: en este parámetro devuelve el número de serie del lector. En caso que la función retorne error devuelve la string vacía.

Valores Retornados:

Retorna PROX_USB_OK (0) si encuentra un lector de proximidad Prox-USB válido, en caso contrario retorna un código de error *(ver cuadro Códigos de resultado)*.

BuscarSiguienteLector(char* num_serie)

Esta función busca un módulo Prox-USB que sigue al módulo con número de serie enviado como parámetro.

Formato:

unsigned long int BuscarSiguienteLector(char* num_serie)

Parámetros:

char* num_serie: en este parámetro se envía el número de serie del lector a partir del cual se busca el lector que le sigue.

Valores Retornados:

Retorna PROX_USB_OK (0) si encuentra un lector de proximidad Prox-USB válido que sigue al número de serie pasado por parámetro.

LeerTarjeta()

Esta función se utiliza para comprobar si fue leída alguna tarjeta de proximidad. En el caso de que se haya realizado una lectura también retorna el número de la tarjeta.
Una vez que la función retorno estos datos, el módulo queda listo para la próxima lectura. Mientras tiene en memoria un número de tarjeta sin descargar, no sigue leyendo otras tarjetas que se aproximen. A partir de la versión 2.0 que contempla manejo de varios ProxUSB en paralelo, es necesario especificar como primer parámetro el ID del lector del cual se leerá el número de tarjeta.

Antes de utilizar esta función es necesario usar BuscarLector().

Formato:

unsigned long int LeerTarjeta(char* num_serie , char *num_tarjeta)

Parámetros:

char* num_serie : ID del lector sobre el cual se realiza la operación.

char * num_tarjeta: es el número de la tarjeta que se aproximó al lector.

Valores Retornados:

PROX_USB_OK (0) si la operación se realizó correctamente pero no hubo lectura.

PROX_USB_FICHADA_OK (1) si hubo una lectura de tarjeta correcta. En este caso el parámetro **num_tarjeta** devuelve una cadena de 10 caracteres numéricos con la información de la tarjeta. Si el número es menor a diez dígitos se completa con ceros a la izquierda. La cadena retornada tiene la siguiente estructura:

```
FFFFFNNNNN
  |   |____ NNNNN = Card Number (se completa con ceros a la izquierda)
  |
  |________ FFFFF = Facility Code (se completa con ceros a la izquierda)
```

PROX_USB_ERROR_PARIDADFICHADA(2) si hubo algun error al validar la paridad de la información obtenida.

Nota:

*La forma normal de implementar una aplicación que requiere la lectura de una tarjeta es armar un loop donde se invoca a la función LeerTarjeta() y monitorear el valor retornado hasta que sea (1). Cuando este es el caso procesar el valor de la tarjeta retornado en **num_tarjeta.***

LeerFechaHora()

Cada dispositivo PROX-USB cuenta con un reloj interno que funciona como reloj y calendario. Esta función retorna la fecha y hora del reloj del módulo. Si la operación se realizó exitosamente la función retorna 0, en caso contrario retorna un código de error. A partir de la versión 2.0 que contempla manejo de varios ProxUSB en paralelo, es necesario especificar como primer parámetro el ID del lector del cual se leerá el número de tarjeta.

Formato:

unsigned long int LeerFechaHora(char* num_serie , unsigned short *dia, unsigned short *mes, unsigned short *ano, unsigned short *hora, unsigned short * minutos, unsigned short *segundos)

Parámetros:

char* num_serie : ID del lector sobre el cual se realiza la operación.

unsigned short *dia: en este parámetro se retorna un valor entre 1 y 31 que indica el día del mes que tiene el calendario del reloj.

unsigned short *mes: este parámetro retorna valores entre 1 y 12, el valor que tome indica el mes del año que tiene el calendario del reloj.

unsigned short *ano: este parámetro retorna valores entre 2000 y 2099, indica el año que tiene el calendario del reloj.

unsigned short *hora: retorna la hora que se encuentra configurada en el reloj. Los valores posibles son entre 0 y 23.

unsigned short *minutos: retorna los minutos que tiene configurados el reloj. Los valores posibles son entre 0 y 59.

unsigned short *segundos: retorna los segundos actuales que han pasado en el relos. Los valores posibles son entre 0 y 59.

Valores Retornados:

Retorna PROX_USB_OK (0) si encuentra un módulo Prox-USB conectado, en caso contrario retorna un código de errror *(ver cuadro Códigos de resultado)*.

GrabarFechaHora()

Cada módulo PROX-USB cuenta con un reloj calendario interno que mantiene la fecha y hora. Para configurar la hora y fecha actual se puede usar la función: GrabarFechaHora, indicando en sus parámetros día, mes, año, hora, minutos y segundos. Además para modificar el horario del reloj es necesario pasarle como parámetro a esta función el password de habilitación. A partir de ese momento el reloj interno de la llave mantiene la fecha y hora actualizada aunque se apague la PC. Para leer la hora y fecha del reloj utilice la función LeerFechaHora. Desde la versión 2.0, que contempla manejo de varios ProxUSB en paralelo, es necesario especificar como primer parámetro el ID del lector del cual se leerá el número de tarjeta.

Formato:

unsigned long int GrabarFechaHora(char* num_serie , unsigned short *dia, unsigned short *mes, unsigned short *anio, unsigned short *hora, unsigned short *min, unsigned short *seg, char *password)

Parámetros:

char* num_serie : ID del lector sobre el cual se realiza la operación.

unsigned long *dia: en este parámetro debe indicarse el día a grabar, debe ser un valor entre 1y 31.

unsigned long *mes: en este parámetro debe indicarse el mes a grabar, debe ser un valor entre 1 y 12.

unsigned long *anio: en este parámetro debe indicarse el ano a grabar, se almacenan los dos últimos dígitos del año solamente. El valor debe pertenecer al rango: 0 - 99.

unsigned long *hora: en este parámetro debe indicarse la hora a grabar, debe ser un valor entre 0 y 24.

unsigned long *min: en este parámetro debe indicarse los minutos a grabar, debe ser un valor entre 0 y 59.

unsigned long *seg: en este parámetro debe indicarse los segundos a grabar, debe ser un valor entre 0 y 59.

unsigned long *password: en este parámetro indique el password del lector de proximidad. Este parámetro solo es requerido para modificar la fecha y hora del lector.

Valores Retornados:

Retorna PROX_USB_OK (0) si la operación se pudo completar correctamente, en caso contrario retorna un código de error *(ver cuadro Códigos de resultado)*.

LeerMemoria()

El dispositivo PROX-USB cuenta con un bloque de memoria no volátil donde el programador puede almacenar información que requiera posteriormente. La función LeerMemoria() permite leer la información almacenada en esta memoria. La cadena leída se retorna en una string la cual debe ser creada antes de invocar a la función. A partir de la versión 2.0 que contempla manejo de varios ProxUSB en paralelo, es necesario especificar como primer parámetro el ID del lector del cual se leerá el número de tarjeta.

Formato:

unsigned long int LeerMemoria(char* num_serie , unsigned long inicio, unsigned long cantidad, char *cadena)

Parámetros:

char* num_serie : ID del lector sobre el cual se realiza la operación.

unsigned long inicio: número de byte a partir del cual se comenzará a leer la memoria.

unsigned long cantidad: cantidad de bytes a leer. La suma entre el byte de inicio y cantidad no debe ser mayor a la memoria disponible del módulo.

Valores Retornados:

PROX_USB_OK (0) si se ha podido realizar la operación de lectura correctamente, de lo contrario retorna un código de error *(ver cuadro Códigos de resultado)*.

GrabarMemoria()

Esta función permite grabar strings en la memoria no volátil del lector. Se puede usar la función LeerMemoria para recuperar las cadenas almacenadas. A partir de la versión 2.0,

que contempla manejo de varios ProxUSB en paralelo, es necesario especificar como primer parámetro el ID del lector del cual se leerá el número de tarjeta.

Formato:

unsigned long int GrabarMemoria(char* num_serie , unsigned long inicio, unsigned long cantidad, char* cadena)

Parámetros:

char* num_serie : ID del lector sobre el cual se realiza la operación.

unsigned long inicio: número de byte donde se comenzará a escribir la memoriaa.

unsigned long cantidad: longitud de la cadena expresada en cantidad de caracteres.

char *cadena: puntero a los datos de la cadena a grabar.

Códigos de resultado

Esta página contiene los códigos de error posible que retornan las funciones exportadas por la librería Prox-USB.DLL:

Nombre	Valor	Descripción
PROX_USB_OK	0	No hubo errores.
PROX_USB_ERROR_ESCRITURA	1	Error al escribir la información en el dispositivo.
PROX_USB_ERROR_LECTURA	2	Error al leer el dispositivo.
PROX_USB_ERROR	3	Error indefinido.
PROX_USB_ERROR_MODELO	4	El dispositivo conectado no es una placa PROX-USB.
PROX_USB_ERROR_SINRELOJ	5	El dispositivo detectado carece de reloj de tiempo real (RTC) o el reloj no está funcionando correctamente.
PROX_USB_ERROR_SINLECTOR	6	El dispositivo detectado no soporta el manejo de lectores de tarjetas RFID.
PROX_USB_FICHADA_OK	1	Se ha descargado la fichada correctamente.
PROX_USB_ERROR_PARIDADFICHADA	2	Se han producido errores al validar la paridad de la fichada.
PROX_USB_ERROR_OUTOFRANGE	7	La lectura/escritura de memoria no se pudo realizar porque los límites están fuera de rango.

Especificaciones técnicas

Parámetro	Detalle
Dimensiones de la placa	67 x 67 mm de lado / altura 15 mm
Alimentación	5V CC (se toma de la propia interfaz USB)
Consumo	20 ma sin lector (*)
Interfaz a PC	USB v1.0
Interfaz al lector de RFID	Wiegand
Protocolo	26 bits Standard
Reloj calendario	RTC precisión +/- 3 minutos por mes
Memoria no volátil	EEPROM serial 4K bytes
Buzzer	Piozoeléctrico de alto rendimiento
Indicadores luminosos	Blanco: funcionamiento normal Azul: tarjeta leída

*(*) La alimentación del lector de tarjetas se toma de la propia interfaz USB.
Es importante contemplar que el consumo del lector no supere los 80 ma.
También se debe verificar que la salida USB utilizada pueda suministrar la corriente necesaria para alimentar la placa ProxUSB más el lector de tarjetas.*